

On the Optimization of Visualizations of Complex Phenomena

Donald House*
Texas A&M University

Alethea Bair**
Texas A&M University

Colin Ware***
University of New Hampshire

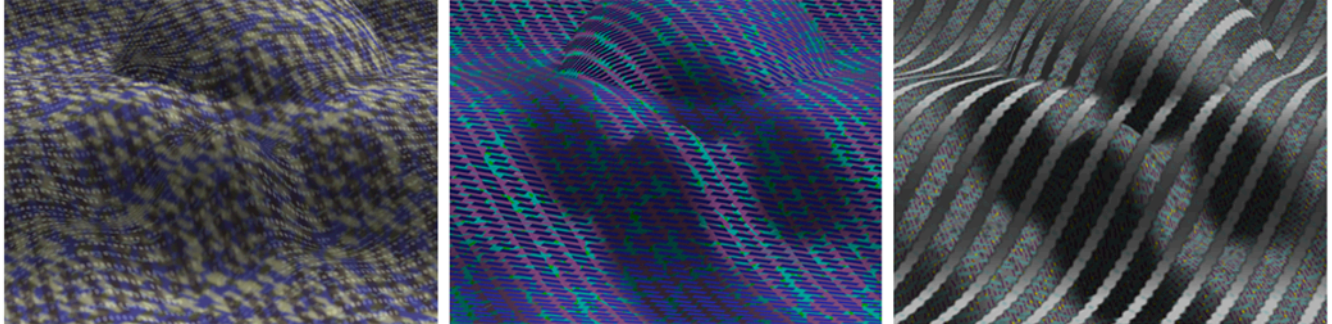


Figure 1. Experimentally determined highly rated solutions to layered surface texturing problem

ABSTRACT

The problem of perceptually optimizing complex visualizations is a difficult one, involving perceptual as well as aesthetic issues. In our experience, controlled experiments are quite limited in their ability to uncover interrelationships among visualization parameters, and thus may not be the most useful way to develop rules-of-thumb or theory to guide the production of high-quality visualizations. In this paper, we propose a new experimental approach to optimizing visualization quality that integrates some of the strong points of controlled experiments with methods more suited to investigating complex highly-coupled phenomena. We use human-in-the-loop experiments to search through visualization parameter space, generating large databases of rated visualization solutions. This is followed by data mining to extract results such as exemplar visualizations, guidelines for producing visualizations, and hypotheses about strategies leading to strong visualizations. The approach can easily address both perceptual and aesthetic concerns, and can handle complex parameter interactions. We suggest a genetic algorithm as a valuable way of guiding the human-in-the-loop search through visualization parameter space. We describe our methods for using clustering, histogramming, principal component analysis, and neural networks for data mining. The experimental approach is illustrated with a study of the problem of optimal texturing for viewing layered surfaces so that both surfaces are maximally observable.

CR Categories and Subject Descriptors: I.3.3 [Computer Graphics]: Picture/Image Generation; I.6.6 [Simulation and Modeling]: Simulation Output Analysis

Additional Keywords: perception, visualization evaluation, layered surfaces, genetic algorithm, data mining, principal component analysis, neural networks

* e-mail: house@viz.tamu.edu

** e-mail: abair@viz.tamu.edu

*** e-mail: cware@ccom.unh.edu

1 INTRODUCTION

There is a strong desire in the visualization community to develop approaches to evaluating the quality and effectiveness of visualization methods. We want to be able to say that our new approach is “good,” but we are immediately confronted with the question: “What do we mean by good?” This is a fundamental problem in all design fields. No matter how good a design may appear, one can always imagine one that is better. Hence, the question of goodness is in one sense a relative question: “Is this better than previous ways of doing this?” At the same time, there are clearly absolute criteria for evaluating goodness, such as: “Can I see what I am supposed to see?” For example, Figure 1 shows three very different experimentally determined good solutions to the problem of optimally texturing layered surfaces for maximum visibility (when viewed in stereo and with motion cues). Each has unique characteristics that make it successful, so the question of which one to use is difficult to answer. Compounding the difficulty is that these relative and absolute questions tend to be answered based on both perceptual and aesthetic criteria.

Perception provides a sensible order to what we see, and aesthetics govern our receptiveness to our perceptions. For example, one of the usually unstated, but very obvious, draws to the *Line Integral Convolution* method for flow visualization [2, 27] is the beauty of the images it creates. This outweighs LIC’s perceptual problems, such as difficulty of seeing direction and speed of flow, and has made it the focus of considerable research, some of which is directed towards solving its inherent problems [14, 20, 30].

In most visualization research, implicit “expert” evaluation is used to determine visualization quality. The researchers themselves evaluate their results and tune their methods to their own satisfaction. Then, they rely on peer evaluation for critique, and eventually to accept or reject the method for publication. In the early stages of the development of a new method, this approach is quite effective, as evaluations do not have to be made at a fine level. However, it can become quite limiting as a method is further refined, and relative judgments become more subtle. Ultimately, expert evaluation is inherently problematic, since

experts often disagree, and their judgments can be colored or limited by their own personal biases or experience.

Keefe et al. [15] found that an artist could produce stylistically interesting and innovative visualizations if they were part of a collaborative team involved in the early stages of the scientific process, and ideally were given access to sophisticated production tools. There is obviously a place for creative design for new problem domains before the inevitable forces for standardization become dominant. A striking, although expected feature of artist's productions, is that one artist's rendition is typically radically different from another and the extent to which general design principles can be extracted from such an effort is unknown.

A recent trend, especially in Information Visualization, has been the employment of user studies (see for example [3, 18, 26]). Plaisant [21] reports that such studies fall into four categories: controlled experiments comparing design elements, usability evaluation of a tool, controlled experiments comparing two or more tools, and case studies of tools. Of these categories, only the first – controlled experiments comparing design elements – is especially useful in developing the underlying visualization methodologies. The others are suited for examining fully developed tools, not the methods themselves. Further, while controlled experiments are clearly applicable to the development of methodologies, they are quite limited in the complexity of phenomena they are capable of addressing.

In a controlled experiment, the object is to hold all experimental variables constant across all trials, except for the one or two that are the subject of study. These variables are modified across trials and objective evaluations are obtained by asking a human subject to perform some task dependent upon understanding the visualization. For example, in a landmark user study in Scientific Visualization, Laidaw et al. [16] compared five specific visualization techniques for flow visualization, including the use of line integral convolution and arrow grids. The results were very revealing in that they showed, for example, that LIC is not very effective in supporting the task of judging advection pathways. However, for each of the five techniques, a particular variant and parameterization was chosen, thus the experiment tells us little about the general properties of a good solution.

We would argue that, in general, the perceptual and aesthetic issues involved in producing an effective visualization are too complex to be fully explored by controlled experiments. It has been our observation that the parameters governing the quality of any visualization are highly coupled or interrelated. Thus, when we hold a set of parameters fixed, we place ourselves at a unique point in parameter space, and variations of selected parameters most likely give us information only about perturbations from that point. If instead we started at a different point in parameter space, the same modifications of parameters might yield very different results. For example, there have been some parametric studies of the way we perceive curved surfaces [5, 19], but as these studies looked at only one or two variables it is difficult to infer general rules. In other words, the validity of controlled experiments depends heavily on an assumed decoupling of the effects of the underlying variables controlling the visualization. If variables interact then the number of measurements required is exponential in the number of display parameters.

To illustrate this point, suppose there are 20 parameters relevant to optimizing a particular visualization and each is given a modest 4 levels. The result is then 4^{20} conditions. Supposing that each requires 20 settings by each subject to obtain a reliable estimate of error, then the result is a need for 20 billion separate measurements for a full parametric study. If each measurement

took 5 seconds it would take more than 10^{14} seconds to carry out the experiment. This is more than forty-thousand lifetimes.

Here, we propose a new approach to optimizing the effectiveness of visualizations. The approach derives from our point of view that, at some level, visualization is a design field dealing with complex phenomena for which there is no single “best” design. It is capable of addressing both the perceptual and the aesthetic issues underlying the notion of high quality in visualization.

Our method is a two-stage process. The first stage engages subjects in a human-in-the-loop data gathering process that has some of the aspects of a controlled experiment, while providing a much greater capability of exploring the space of visualization parameters. Data gathering is followed by data mining, which attempts to understand the structure of the visualization parameter space in order to discover information useful to the visualization designer. The rest of this paper elaborates this process in detail, and provides an illustrative example using the problem of visualizing layered surfaces.

2 METHODOLOGY

The data-gathering and data-mining phases of our method for optimizing visualizations are described below.

2.1 Data gathering

In the data-gathering phase we do a human-in-the-loop guided search through the visualization parameter space, producing a large set of evaluated visualizations. This is shown in block diagram form in Figure 2. This search engages human subjects (who can be either expert or naïve) who rate solutions on grounds that the experimenter considers appropriate to the problem. These grounds can include both perceptual and aesthetic concerns. In broad outline the data-gathering phase consists of the following steps:

- 1) choosing a visualization method for a specific class of visualization problems,
- 2) developing a parameterization of the method suitable to the problem, so that a vector of parameters, together with the chosen method, controls the visual representation,
- 3) running a set of experiments designed to search the parameter space, progressively producing an improved set of solutions, guided by user evaluations,
- 4) building a database of the rated visualization solutions visited during the experiments.

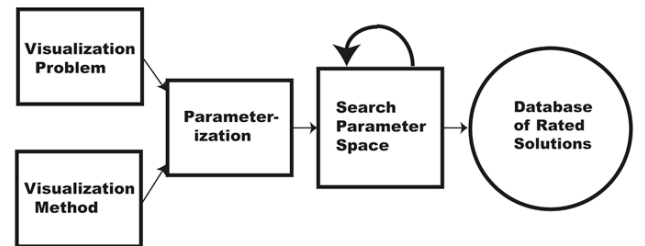


Figure 2. Data-gathering phase of human-in-the-loop exploration

We have found that a genetic algorithm [7], using genes corresponding to the visualization parameters, is a good choice for guiding the parameter space search. This follows work by Dawkins [6], Sims [25] and Greenfield [8] who coupled image generation with user feedback in the context of a genetic algorithm, usually to control aesthetic content of images. It also

follows the notable study by He et al [9] who used this approach to optimize parameters controlling the transfer function used in rendering volume data sets.

The genetic algorithm maintains a relatively small population or generation. Since each genome encodes a complete visualization, it is straightforward to produce the visualization for each member of a generation and to employ a human subject to provide an evaluation, rating the solution's quality. This evaluation can range from a subjective judgement to a measure of performance on a visual task. The rated solutions are used to guide the progress of the algorithm and are also stored in the database. The process is capable of methodically producing a large set of generally improving pre-evaluated visualization solutions.

2.2 Data mining

After the data-gathering phase is completed, we have a database of rated solutions sampling the visualization parameter space, and we can proceed with the data-mining phase that is depicted in Figure 3. The goal of this phase is to glean information about what leads to effective visualizations and what detracts. The ultimate object would be to gain enough of an understanding of the structure of the visualization parameter space to yield formal theories of visual mappings that create effective visualizations. Although a single study is not likely to lead to such strong results, we would at the very least be able to identify a set of exemplary visualizations for a specific problem. Beyond this, it is likely that we would be able to specify sets of helpful default parameter settings, or better, design guidelines that lead to strong visualizations. Ranging from the least to the most general, results could be:

- 1) descriptions of the characteristics of an effective visualization for a specific problem,
- 2) a description of those characteristics for a generalized problem,
- 3) formal sets of characteristics that can be applied automatically or algorithmically,
- 4) informal theory of visual mappings that create an effective visualization,
- 5) formal theory of effective visual mappings.

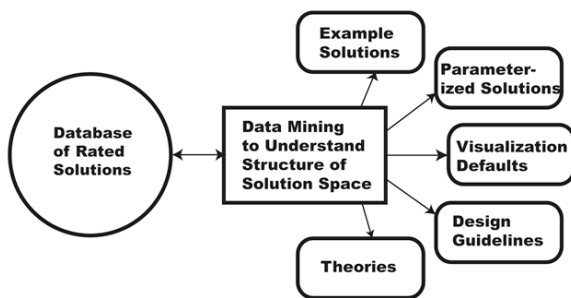


Figure 3. Data-mining phase of human-in-the-loop exploration

Although this is described as a two-phase process, the technique can feed back on itself, using results from data mining to form hypotheses that can be tested by further experimentation. Thus, a series of experiments can be generated to further test and refine results. For example, new experiments could start with a hypothesis, gleaned from the data mining, about how particular parameter settings affect visualization quality, and, starting with exemplar solutions also from the data mining, test this hypothesis

by methodically varying these parameters from their base values in the exemplars.

We have experimented with a number of data-mining methods for extracting information from the database. The most straightforward approach is clustering of highly rated visualizations [17], with cluster membership determined by Euclidian distance in parameter space. Each cluster then consists of a number of solutions that share high ratings, and can be represented by an exemplar taken from the center of the cluster. Visual inspection of these exemplars can lead to descriptions of their salient features, which gives an idea of the variety of parameter setting strategies that lead to good visualizations. Examination of interrelationships across clusters can lead to the development of rules, procedures or hypotheses for developing successful visualizations. These can be tested experimentally, with the goal of developing concrete guidelines or even theories.

Hypotheses coming from cluster inspection can be subjected to more rigorous statistical analyses on the database. In our work, we construct a non-linear function of parameters that is meant to test the hypothesis, apply this function to the parameter sets in the database, and build a histogram of the result. Building the histogram of the same function applied to a random population gives us our null hypothesis. Comparing this distribution to histograms of visualizations receiving very high scores gives us a simple visual method of looking for significant deviations from the null hypothesis. More sophisticated approaches can also be used, and we are experimenting with a number of these.

Principal component analysis of visualizations with high scores allows us to find vectors in parameter space along which the maximum variance in parameters occurs. If we pick a cluster center as the origin of a coordinate system in parameter space, then the principal components give coordinate directions along which we can expect to vary visualization parameters without degrading the quality of the visualization. This can be very useful in identifying “free parameters” that we can adjust to encode information in the visualization.

We are also experimenting with neural network analysis, training a network on the database to produce scores from parameter sets that match the scores obtained during the experiments. What this gives us is a mapping from the parameter space to ratings – i.e. the neural network is a black-box function that takes parameter settings as its input and outputs corresponding ratings. Ideally, we would like to be able to input ratings and receive sets of parameters as outputs, but the function implemented by a neural network is only onto, not one-to-one, and therefore is not uniquely invertible. Further, since the units of the neural network use nonlinear transfer functions, examination of the network weights to determine simple relationships among parameters and ratings is problematic.

3 LAYERED SURFACE TEXTURING EXAMPLE

In order to illustrate our method more clearly, we offer the following example of a study of texture mapping to enhance the visualization of layered surfaces. This study is presented in preliminary form in [10], and with full details and results of recent work in [1].

One of the most important yet perceptually difficult problems in data visualization is that of displaying one surface overlaying another. Just a few of the numerous applications of layered surface visualization include: medical imaging – to see the shape of different tissues overlying each other; geological applications – to see how geological layers are situated with respect to each

other; and oceanography – to see how isosurfaces defined by (for example) temperature change relate to the seabed topography.

What makes layered surface visualization such a difficult perceptual problem is the visual confounding between the images of the two surfaces. For example, shape-from-shading information can be impossible to perceptually separate. When encountering layered surfaces in the real world – like viewing a scene through a screen of shrubbery – we have the advantage of highly coupled vergence and accommodation cues. As our eyes bring one surface into binocular registration and sharp focus, the other surface becomes blurred, thus reducing the confounding of visual cues. On a computer screen, however, both surfaces are presented on the plane of the display surface, so that we lose the perceptual advantages of the three-dimensionality of the real problem. We can help matters by providing some depth information through both stereoscopic viewing and motion parallax [29]. Further, Interrante et al. [11] found that adding distinct partially-transparent textures to the layered surfaces can help to distinguish them. Nevertheless, even with all of these perceptual aids, there is still a strong tendency to visual confusion.

3.1 The visualization problem

We defined the visualization problem to be: how to choose pairs of draped textures for two surfaces so that, when the surfaces are overlaid and viewed in stereo and in motion, they optimally reveal the shapes of both surfaces and do not perceptually interfere with each other. Because textures can be arbitrarily complex, this is not an easy problem to solve. It can take tens of parameters to define a single complex texture with a reasonable set of texture elements and color components. Further, there is the issue of how the textures should be oriented with respect to the viewpoint and the surface topography. Due to the number of parameters, it is difficult to see how much progress can be made on this problem simply using controlled studies.

3.2 The visualization method

The method that we chose for studying the layered surface problem was to fix viewing and surface parameters, while varying the textures applied to the two surfaces. Thus, the study has elements of a controlled study, but the experimental variables are extremely complex. The scene consisted of the overlay of the two surfaces shown in Figure 4. The bottom surface is a flat plane with hills in the center defined via a Gabor function. The top surface has a long period sinusoidal wave whose front is nearly perpendicular to the viewing direction, and a large dome-like structure. The planes of the two surfaces are parallel, tilted away from the camera by 30 degrees and separated by slightly more than the height of the tallest feature on the bottom surface (to avoid interpenetration of the surfaces). The scene is lit using a single parallel light source with direction vector $\langle 1, 1, 1 \rangle$, and shading is done using a simple 70% lambertian + 30% ambient shader (i.e. without specular highlights). The surfaces are viewed in stereo, and rocked about the center vertical screen axis so that both stereo and motion cues are available to resolve depth.

It has been shown that textures grown to conform to features of a surface are very powerful in conveying surface shape in monocularly viewed static images (see especially Kim et al. [13]). However, in our study we consider only simple draped nonconformal textures. We are aware of no evidence that conformal textures improve on simple draped textures under stereo viewing. Further, the simplicity and ubiquity of use of draped textures argue for their continued study.

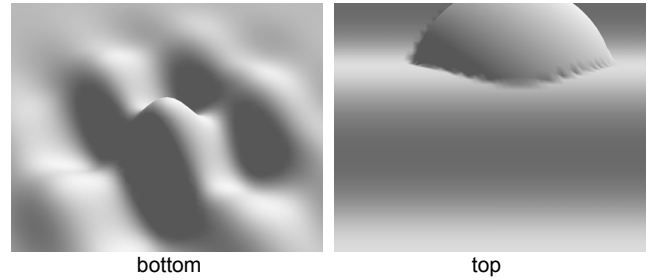


Figure 4. Surfaces used in the layered surface experiment

3.3 Problem parameterization

We developed a parameterized texture space that would allow us to test a number of texture attributes that we felt might bear on the layered surface problem. Refer to [10] for complete details on the texture generation method. The overall texture attributes that we decided to parameterize were: 1) orientation, 2) foreground transparency, 3) density of pattern, 4) regularity of pattern (i.e. structured vs. random), 5) softness (i.e. soft vs. hard edges), and 6) background color. The parameterized attributes of individual texture elements making up the pattern were: 1) transparency, 2) size, 3) linearity (i.e. long vs. short strokes vs. dots), 4) orientation, and 5) color. Textures were algorithmically constructed from the parameters by first building a background layer and then drawing in three separate sets of texture elements to create a texture tile. The background layer is of a constant HSV α color, and the HSV α texture elements consist of one set of dots and two sets of linear strokes. Using this approach, results can vary across a range from a fully transparent background with opaque texture elements (giving the illusion of texture elements floating in space) to a translucent background with translucent texture elements (giving the illusion of a continuous textured surface).

Using this structure we parameterized a pair of textures in a vector of 122 elements. Figure 5 shows two foreground/background texture pairs generated in this manner.

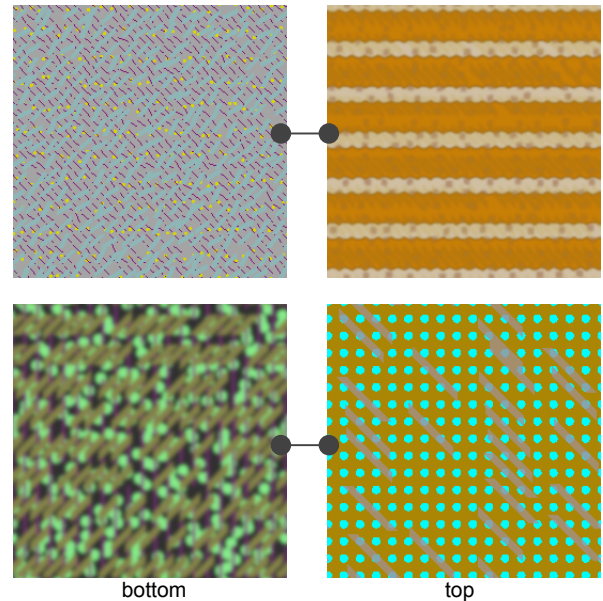


Figure 5. Example texture tiles

In the figure, texture pairs are arranged horizontally. The front textures have transparencies, so for illustration purposes they are shown composited over orange (i.e. orange regions are transparent). Seven parameters per texture control overall appearance, and 18 parameters control each of the three sets of texture elements per texture.

3.4 Parameter space search

The visualization parameter space was searched using experimental trials, driven by an underlying genetic algorithm, and conducted in the following way. For each presentation, a subject was shown the surfaces from Figure 4, textured according to a trial set of parameters, and asked to qualitatively evaluate their ability to clearly see the features of both the bottom and top surfaces. To make sure that subjects understood what they were to be looking for, at the start of each session the subject was shown the surfaces with hand-built textures that clearly showed both surfaces. The rating scale was 0-9, and input was made using a single numeric key press on a standard keyboard. These scores were recorded with each texture pair, and were used to determine fitness in the genetic algorithm. For the genetic algorithm, a single generation consisted of 40 texture pairs. Each of these pairs was presented in sequence until all were evaluated. Once a full generation was evaluated, breeding between textures was done using a two-point crossover approach, with the probability of a texture pair being selected for breeding determined by the ratings. Our study involved five subjects, each completing three full experimental trials of 15 generations. To reduce the effect of fatigue, subjects were able to save results at the end of any generation, and continue again at a later time. One complete experimental trial took about three working hours. Each trial successfully converged to produce a generation with a high percentage of the textures receiving high ratings. Figure 6 shows two image snapshots, with different texture pairs on the bottom and top surfaces, taken from two different points in our experimental trials.

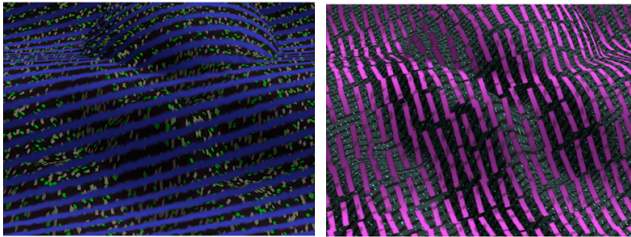


Figure 6. Example presentations in layered surface experiment

3.5 Experimental database

At the end of each generation, all texture pairs and their ratings were stored in an output file. A merge of all files from all experiments gave us a database of over 9000 evaluated surface texture pairs. In this database about 10% of the texture pairs have very low ratings (0 or 1), while about 20% have very high ratings (8 or 9). Figure 7 provides a comparison of the expected distribution of ratings given a completely random data set constructed from the ratings of the first generations only (dashed line) vs. the distribution of ratings obtained over all trials (solid line). It is clear that the algorithm devotes a significant amount of time to exploring fruitful areas of the parameter space, while much more lightly sampling poorer areas.

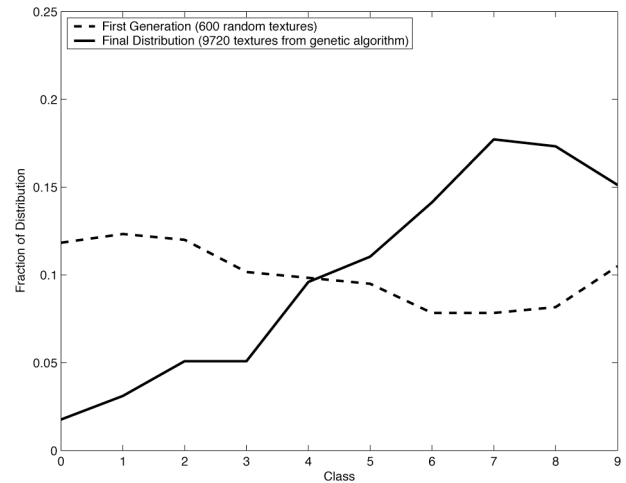


Figure 7. Ratings in experimental database vs. random set

3.6 Data mining

We have experimented with a number of methods for gleaning information from our experimental database. These are briefly surveyed below. A fuller review of results is provided in [1].

3.6.1 cluster analysis

Our earliest data-mining attempt was using cluster analysis. Figure 1 (on the first page) shows cluster medians from three of the many clusters found in the database. Clusters were formed using a hierarchical-nearest-neighbors approach [12], so they are not typically spherical but can have elongated shapes in parameter space. In theory, this could lead to cluster elements having very different visual characteristics, but in practice we found that all visualizations within one cluster are visually identifiable as belonging to one “family.” What is immediately apparent from examination of Figure 1 is that these clusters are quite diverse in their structure and appearance. We found that in most, the percentage opacity of the top surface texture was a key factor, with texture elements being fully opaque and the background being fully transparent, as in the center and right image in Figure 1. However, there were several good solutions having a milky translucent surface scattered with small texture elements, as in the left image in Figure 1. Other indicators are that many, but not all, of the good solutions have texture components that differ greatly in size between the foreground and the background. Less obvious, but still apparent is a tendency for more structure on the top surface and a more random appearance on the bottom surface.

3.6.2 histogram analysis

Visual analysis of clusters can lead us to hypotheses about what attributes of texture pairs lead to effective layered surface visualizations. We have found that histogram analysis is an effective way to test simple hypotheses involving non-linear combinations of multiple parameters, and to develop quantitative guidelines for producing strong texture pairs.

For example, cluster analysis tells us that there should be opaque areas on the top surface, but by histogram analysis we can get a feeling for how much coverage there should be. We start by building a measure of top surface coverage, expressed in the form of a function of texture parameters. We then histogram this function for a completely random data set, and compare it with the histogram of texture pairs scored highly (8 or 9). Figure 8 shows such a comparison. The dashed line represents coverage for the random set and the solid line the coverage for the highly rated

set. Error bars indicate 95% confidence intervals. What we can learn from the plot is that many more good top textures have coverage between 20% and 70% than random, and many fewer have coverage between 70% and 90%. We can infer a preference for about 30% coverage, as this is where the peak positive difference occurs. Coverage values below 10% and above 90% are not graphed, since our coverage measure is inaccurate there.

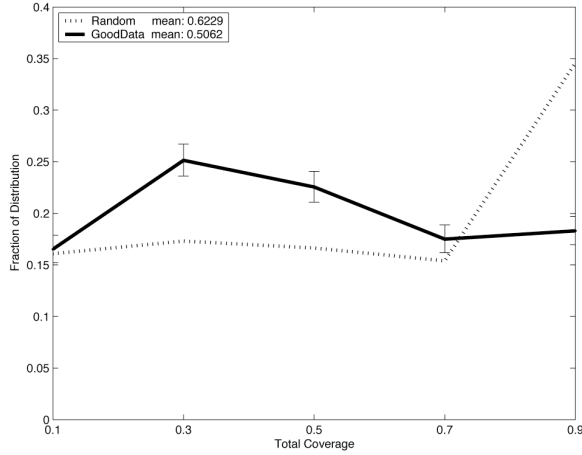


Figure 8. Top surface coverage

Another interesting result is shown in Figure 9. Here we plot the difference between jitter of texture elements on the top surface and jitter on the bottom for highly rated texture pairs. We see that significantly fewer highly rated texture pairs have zero difference in jitter than in the random population, and that the distribution is skewed showing that highly rated textures tend to have more jitter on the bottom surface than on the top. Thus we can conclude that variation in regularity across the two surfaces is helpful, and that making the top surface more regular than the bottom is preferred.

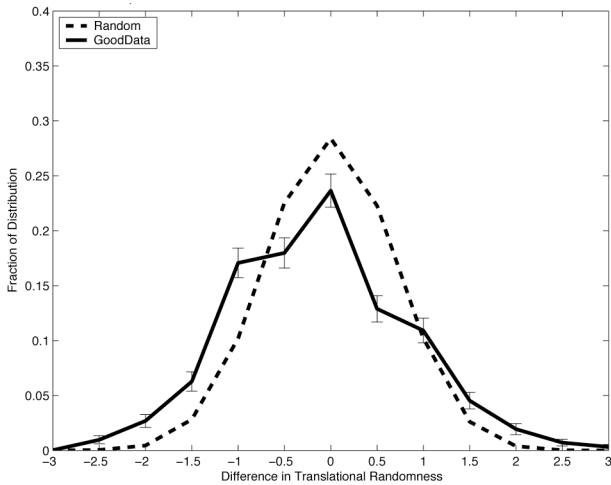


Figure 9. Jitter difference across surfaces

Finally, it is instructive to take a look at color differences across the surfaces. Figure 10 top shows that highly rated textures exhibit no more difference in hue than in the random set. This leads us to conclude that hue may be available as a “free” parameter, useful for encoding information without interfering with the visual separation of the layers. Figure 10 bottom indicates that the maximal value differences between salient features across the

surfaces should be about 40%, with a preference for having the bottom surface at higher value.

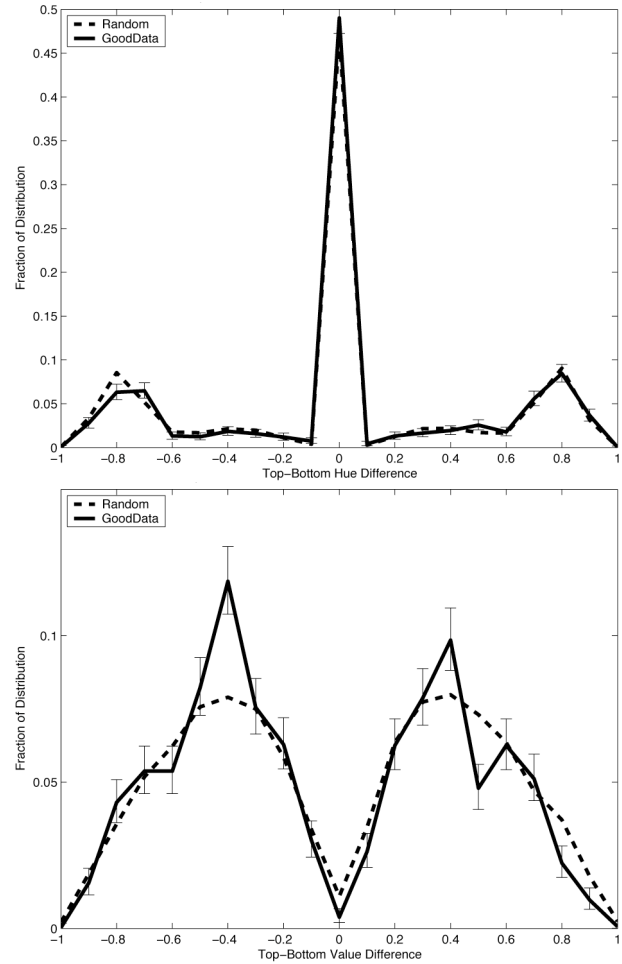


Figure 10. Color differences across surfaces

3.6.3 principal component analysis

We use principal component analysis of clusters of highly rated solutions to discover directions in parameter space along which visualizations may be varied without disturbing visualization quality. The middle image shows the median texture in a cluster. To its left and right are new visualizations generated by following the first principle component in both directions from the cluster mean. The changes in orientation, color, and texture granularity do not degrade the visualization. The features represented by the principal component vectors can therefore behave as free parameters when constructing good textures. Unfortunately this method does not provide specific rules for making good textures, since variation across the parameter space is ignored. However, it does give an indication of which parameters are more important.

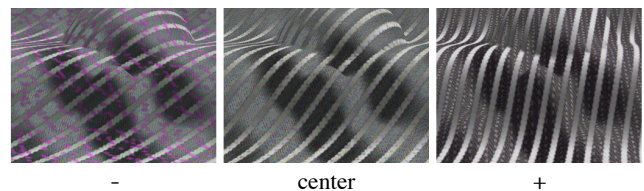


Figure 11. Variants following first principal component

In analyzing the principal components corresponding to the largest eigenvalues, several trends were apparent. Comparable parameters always varied more on the top surface than the bottom surface. Also, with the exception of transparency, the comparable parameters on the surface background varied more than those for the texture elements. This implies that careful choice of settings for bottom surface characteristics are more important than those for the top, and that texture feature characteristics are more important than the texture background. Reflecting our results from histogram analysis, hue and saturation variables had more variation than value, and certain values of the value parameter are likely to be much better than others for creating good textures. Interestingly, parameters encoding the shape of features, such as the number of rows and columns in the grid, size and shape of the elements, and randomness of the features, always varied less than the color parameters. This indicates that features must have good placement, size and shape before parameters like color, rotation and filtering can have much of an effect on visualization quality.

3.6.4 neural network analysis

Neural network analysis allows us to move away from cluster centers and consider the structure of the entire solution space. We built a 2-layer back-propagation network with 122 inputs corresponding to the parameter space. These are fully connected to 20 hidden units, which in turn are fully connected to 10 output units corresponding to texture ratings on a scale of 0-9.

Thus, each texture is input as a vector of parameters, and the output is a classification (0-9). Using only 20 hidden units provides a large data reduction from the 122 inputs, but the network learned to categorize with reasonable accuracy. Given a network that correctly classifies the textures, understanding the meaning of the weights is difficult. The non-linearity of the network units prevents a simple analysis of weight vectors, however simply looking at which features had large magnitude positive or negative weights leading to an output node proved interesting.

Examining the weights to output unit 9 (most highly rated), we found that the top surface transparency was high, that little low-pass filtering was done, and rotation of the overall texture from horizontal was high. Widely separated small lines were preferred, with little horizontal or rotational jitter but large vertical jitter. Interestingly, a single set of lines was preferred on the top surface, and dots were not typically used. This corresponds with other indications that there should be more structure on the top surface. On the bottom surface, high background value was preferred. In contrast to the top surface, the use of small, randomly placed dots with high value and saturation was indicated.

4 DISCUSSION

One of the interesting aspects of our approach to visualization optimization is the confounding of perceptual and aesthetic issues. We found in our pilot study that it was very difficult to get subjects to respond to textures from the purely objective viewpoint of maximizing visibility. Instead, there was always an instant response to textures deemed “beautiful”, and a strong tendency to score them high, even if they might not work particularly well. Interestingly, we even found this to be true of ourselves when we acted as experimental subjects! Depending upon one’s goals, this can be seen as either a drawback or a strength of our optimization approach. Of course, the use of a more objective evaluation system than we used in the example experiment would help to reduce this problem.

It is clear that a simple genetic algorithm approach to searching the visualization parameter space is not ideal. For the experimental subjects it is a long, slow process to arrive at consistently strong visualizations. One of the frustrations, especially for an expert subject, is the inability to use domain knowledge to directly guide the search process. In the layered texture experiments, we would frequently see a texture that could be improved immensely in obvious ways, but all we could do was to score it and move on. We have two ideas to augment the GA approach, that we feel will help. The first is to implement an “islanding” capability [24] that allows creation of an “island” population of textures, all nearby in parameter space to a texture that the subject finds interesting. The island population could then be evolved by itself or later merged back into the general population. Our second augmentation to the GA would be to provide an interface that would allow direct “tweaking” of the visualization parameters. Tweaked visualizations could then be inserted into the population to affect future evolution.

Although neural network analysis cannot give us a mapping from ratings to parameters, it can be helpful in a number of ways. Most especially, it might be used to numerically estimate gradients in parameter space. We are looking at ways in which this could be coupled back to the data-gathering phase to assist in guiding the search through the visualization parameter space. Thus, results of previous experiments could be used to make new experiments more efficient. The neural network can also be used to more densely populate the database by randomly generating parameter sets, scoring them using the network, and using them to selectively fill in gaps in the database. This is the basis of Craven and Slavik’s Trepan algorithm [4] for generating decision trees from a data set and a trained neural network. These decision trees essentially reinterpret the weights of the neural network as a set of logical expressions involving parameters and their settings, leading to a much more explicit description of the structure of the parameter space.

Data mining is sensitive to choice of parameterization for a particular problem. It is especially important to avoid parameters that act as switches among cases, instead opting for parameters that provide smoother variation. Case switches tend to have very high variance in themselves, and confound the difficulty in evaluating other parameters, whose meanings may change based on the case selected. In the layered surface experiment, we used switches that turned low-pass filtering on or off, and others that controlled the type of random jitter for texture features. These made analysis of the corresponding parameters quite difficult, while providing little benefit. A smoother alternative to switches would be to provide *a priori* probability density functions to bias how parameters are randomly selected. For example, instead of a binary parameter to turn jitter on or off, we could have provided a *pdf* for jitter that would yield 50% jitter parameters of 0, and a uniform distribution of the remaining 50% over the rest of the range. Another simple but useful idea is to always use an odd number of values when parameters take on discrete values. This has the consequence of providing a center value that is in the set.

One unsatisfying aspect of our experimental methodology is that we have not attempted to look at inter and intra subject variability in the evaluation of visualizations. Examining intra subject variability could be easily incorporated into the existing genetic algorithm, by reinserting randomly chosen presentations into the experiment during the scoring process. Inter subject variability could be examined by similar methods but selecting presentations from other subjects. Alternatively or additionally, we could run a set of later trials in which we ask all of our subjects to rescore the same representative set of presentations.

5 CONCLUSION

We have described a new approach to the optimization and evaluation of visualization solutions that is capable of accounting for both perceptual and aesthetic factors affecting visualization quality. We have shown how a genetic algorithm, utilizing human-in-the-loop evaluation, can be a practical way of searching the space of parameterized solutions to a visualization problem, and have proposed a number of data-mining techniques for extracting useful information from the database produced during experimental trials. These include clustering, histogram analysis, principal component analysis and neural networks. Histogramming functions of parameters appears to be particularly useful in exploring parameter interrelationships.

The most challenging and promising part of the application of this new methodology is in developing more powerful data mining techniques. There is a wealth of related literature in far-flung disciplines such as psychology, sociology, economics, and ecology. We feel that exploration of this literature in the context of visualization holds promise of greatly contributing to the development of the field.

We are currently at work incorporating what we have learned from our preliminary study of layered surface texturing into two new studies to be conducted using this new methodology. The first of these will take another look at layered surface texturing, but incorporate a greatly reduced and more carefully designed parameter space (24 vs. 122 parameters), surfaces with fixed but broad spatial frequency content that vary in shape with each trial, and a somewhat more objective evaluation criterion (ease of finding a fixed number of features of varying scale). The second study will examine the problem of fluid flow visualization.

6 ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation ITR's 0326194 and 0324899, the Center for Coastal and Ocean Mapping – University of New Hampshire, the Visualization Laboratory – Texas A&M University, and the Texas A&M Department of Architecture. Finally, we wish to thank the anonymous *Vis2005 Conference* reviewers whose keen observations and guidance have improved this presentation.

REFERENCES

- [1] Bair, A. House, D. Ware, C. Perceptually optimizing textures for layered surfaces, *Proceedings of Symposium on Applied Perception in Graphics and Visualization (APGV05)*, to appear.
- [2] Cabral, B. and Leedom, L. Imaging vector fields using line integral convolution, *Computer Graphics (Proceedings of SIGGRAPH 93)*, 263-270, 1993.
- [3] Chen, C. and Czerwinski, M. P. Empirical evaluation of information visualizations: an introduction, *Int. J. Human-Computer Studies*, 53, 631-635, 2000.
- [4] Craven, M. Shavlik, J. Using neural networks for data mining, *Future Generation Computer Systems*, 13, 211-229, (1997).
- [5] Cumming, B.G. Johnston E.B. and Parker, A.J. Effects of different texture cues on curved surfaces viewed stereoscopically, *Vision Research*, 33(56): 827-838, 1993.
- [6] Dawkins, R. *The Blind Watchmaker*, Harlow Logman, 1986.
- [7] Fogel, D. B. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, NJ, 2nd edition, 1999.
- [8] Greenfield, G. Color dependent computational aesthetics for evolving expressions, *Bridges: Mathematical Connections in Art, Music, and Science; Conference Proceedings*, 9-16, 2002.
- [9] He, T. Hong, L. Kaufman, A. and Pfister, H. Generation of transfer functions with stochastic search techniques, *Proceedings of IEEE Visualization 96*, 227-234, 1996.
- [10] House, D. and Ware, C. A method for the perceptual optimization of complex visualizations, *Proceedings of Advanced Visual Interfaces (AVI'02)*, 148-155, 2002.
- [11] Interrante, V. Fuchs, H. and Pizer, S.M. Conveying shape of smoothly curving transparent surfaces via texture. *IEEE Trans. on Visualization and Computer Graphics*, 3(2) 98-117, 1997.
- [12] Johnson, S. C. Hierarchical clustering schemes, *Psychometrika*, 2, 241-254, 1967.
- [13] Kim, S. Hagh-Shenas, H. and Interrante, V. Conveying shape with texture: experimental investigations of the texture's effects on shape categorization judgments. *IEEE Trans. on Visualization and Computer Graphics*, 10(4) 471-483, 2004.
- [14] Kiu, M-H. and Banks, D. C. Multi-frequency noise for LIC, *Proceedings of IEEE Visualization 96*, 121-126, 1996.
- [15] Keefe, D.F. Karelitz, D.B. Vote, E.L. and Laidlaw, D.H., Artistic collaboration in designing VR visualizations. *IEEE Computer Graphics and Applications* (in press).
- [16] Laidlaw, D. H. Kirby, M. Jackson, C. Davidson, J. S. Miller, T. DaSilva, M. Warren, W. and Tarr, M. Comparing 2D vector field visualization methods: A user study. *Transactions on Visualization and Computer Graphics*, 11(1), 59-70, 2005.
- [17] Marks, J. Andalman, B. Beardsley, P.A., Freeman, W., Gibson, S., Hodgins, J. and Kang, T. Design galleries: a general approach to setting parameters for computer graphics and animation, *Computer Graphics (Proceedings of SIGGRAPH 97)*, 389-400, 1997.
- [18] Morse, E. Lewis, M. and Olsen, K. A. Evaluating visualizations: using a taxonomic guide. *International Journal of Human-Computer Studies*, 53 (5), 637-662, 2000.
- [19] Norman, J.F., Todd, J.T. and Phillips, F. The perception of surface orientation from multiple sources of optical information. *Perception and Psychophysics*, 57(5), 629-636, 1995.
- [20] Okada, A. and Kao, D. L., Enhanced line integral convolution with flow feature detection, *Proceedings of IS&T/SPIE Electronics Imaging 97*, Vol. 3017, 206-217, 1997.
- [21] Plaisant, C. The challenge of information visualization evaluation, *Proceedings of Advanced Visual Interfaces (AVI'04)*, p.109-116, 2004.
- [22] Ramachandran, V. Perceived shape from shading. *Scientific American*, August, 76-780, 1988.
- [23] Rogers, B. and Cagnello, R. Disparity curvature and the perception of three-dimensional surfaces. *Nature* 339, May, 137-139, 1989.
- [24] Ryan C. Niche and species formation in genetic algorithms, in Lance Chambers, Ed., *Practical Handbook of Genetic Algorithms*, Vol. 1, CRC Press, Inc., Boca Raton, 58-73, 1995.
- [25] Sims, K. Artificial evolution for computer graphics, *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25, 319-328, 1991.
- [26] Stasko, J. Catrambone, R. Guzdial, M. and McDonald, K. An evaluation of space-filling information visualizations for depicting hierarchical structures, *International Journal of Human-Computer Studies*, 53 (5), 663-694, 2000.
- [27] Stalling, D. and Hans-Christian Hege, H-C. Fast and resolution independent line integral convolution, *Computer Graphics (Proceedings of SIGGRAPH 95)*, 249-256, 1995.
- [28] Todd, J.T. and Akerstrom, R. Perception of three-dimensional form from patterns of optical texture, *J. Experimental Psychology: Human Perception and Performance*, 13 (2), 242-255, 1987.
- [29] Ware, C. and Frank, G. Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Transactions on Graphics* 15(2): 121-140, 1996.
- [30] Wegenkittl, R. Groller, E. and Purgathofer, W. Animating flow fields: rendering of oriented line integral convolution, *Proceedings of Computer Animation 97*, 15-21, 1997.